MENTATION PAGE

BEFORE COMPLETING FORM

1. REPORT NUMBER #59	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A novel recursive partitioning criterion.		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) M.P. Perrone		8. CONTRACT OR GRANT NUMBER(s) N00014-91-J-1316
9. PERFORMING ORGANIZATION NAME AND ADDRESS Institute for Brain and Neural Systems Brown University Providence, RI 02912		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS N-201-484
11. CONTROLLING OFFICE NAME AND ADDRESS Personnel & Training Research Program Office of Naval Research, Code 442PT Arlington, Virginia 32217		12. REPORT DATE 12/23/92 13. NUMBER OF PAGES 5 pages
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited. Publication in part or in whole is permitted for any purpose of the United States Government.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Published in Proceedings of the International Joint Conference on Neural Networks, volume II, page 989. IEEE 1991.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

CART
Recursive Partioning
Hybrid Networks
Misclassification Matrix

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A data-driven algorithm for partitioning many-class classification problems is presented. The algorithm generates tree-structured hybrid networks with controller nets at tree branches and local expert nets at the leaves. The controller nets recursively partition the feature space according to a novel misclassification minimization rule designed to create groupings of the classes which simlify the classification task. Each local expert is trained only on a subset of the training data corresponding to one of the partitions. The advantae to this approach is that the classification task that each local expert performs is greatly simplified. This simplification helps to avoid the curse of dimensionality and scaling problems by allowing the local expert nets to focus their search for structure in a small portion of the input space.

A Novel Recursive Partitioning Criterion[1]

Michael P. Perrone

Physics Department and Center for Neural Science
Box 1843, Brown University
Providence, RI 02912
Email: mpp@cns.brown.edu

Abstract

A data-driven algorithm for partitioning many-class classification problems is presented. The algorithm generates tree-structured hybrid networks with controller nets at tree branches and local expert nets at the leaves. The controller nets recursively partition the feature space according to a novel misclassification minimization rule designed to create groupings of the classes which simplify the classification task. Each local expert is trained only on a subset of the training data corresponding to one of the partitions. The advantage to this approach is that the classification task that each local expert performs is greatly simplified. This simplification helps to avoid the curse of dimensionality and scaling problems by allowing the local expert nets to focus their search for structure in a small portion of the input space.

Hybrid Neural Networks

Recent convergence theorems for a variety of neural network architectures show that certain neural networks can perform arbitrary functional mappings (See, for example, Barron89, Cybenko89, Hampshire90). These results represent worst case bounds for network performance and can be improved by using data-driven techniques if one can assume the existence of appropriate structure in the data. (See, for example, Bachmann90, Breiman84, Ersoy90, Friedman88, Nowlan90, Reilly88, Reilly87, Sanger90, Sankar91)

One can use hybrid networks to implement data-driven algorithms in a neural network setting. (See Cooper91 for further discussion and references.) The hybrid approach is to divide a large network into many smaller networks — called local experts — depending on the data presented to the algorithm. Each local expert then focuses only on a small task. This division of labor among various networks helps in the poor scaling problems of large networks by reducing the required complexity of each of the individual networks. If each individual network is solving a small portion of the total problem then it will necessarily be less complex. However, the division of the data among several networks may increase the bias of the architecture to the training data and therefore require special care. Consider an

[1] The work on which this article is based was supported in part by the National Science Foundation, the Army Research Office and the Office of Naval Research.

optical character recognition task: A network which identifies all letters will certainly be more complex than a network which only distinguishes between "V" and "U".

This paper presents some results from ongoing research into hybrid network algorithms at the Center for Neural Science and Nestor Inc.

The Partitioning Problem

When designing a hybrid network algorithm, one must decide how the feature space will be divided among the local experts. We call this problem the partitioning problem. One effective approach to the partitioning problem is to have local expert nets competing for regions of the feature space (Nowlan90). Another approach is to dedicate local experts to particularly problematic regions of feature space (Cooper91, Reilly87, Reilly88). The approach we take in this paper is motivated by work by Bachmann (Bachmann90) in which multi-class tasks were partitioned by arbitrary class groupings. In this paper, we present a more systematic approach to forming class groupings which looks for the simplest partition at each step.

The first step in partitoning a particular group of N classes into subgroups is to generate a misclassification matrix from a network which has been trained on the full N class problem. The misclassification matrix element m_{ij} is the empirical probability that the trained network will classify test patterns from class i as belonging to class j. Thus any off-diagonal terms correspond to misclassification and a perfect network would produce a diagonal matrix.

We now define a partition as a grouping of the N class labels into two, non-empty subgroups α and β of labels. Our desire is to find the partition with the simplest decision boundary. One measure of the simplicity of a decision boundary is how often patterns are misclassified across it. Therefore we define an inter-group misclassification measure, $M(\alpha,\beta)$, as follows:

$$M(\alpha,\beta) \equiv \sum_{i\in\alpha, j\in\beta} m_{ij},$$

where m_{ij} is an element from the misclassification matrix and α and β are the subgroups of the partition. (Fig. 1) We now define a good partition as one for which $M(\alpha,\beta)$ is minimized. Thus, a good partition is one for which there is a minimum of misclassification between groups.

For very large problems or for problems with a high percentage of misclassifications, it may be beneficial to include a penalty term for unbalanced partitions. An unbalanced partition is one for which one of the subgroups has many more classes than the other. (Fig. 1) For such a partition it is possible that minimizing an unpenalized misclassification measure does not represent a good partition. Therefore, one may choose to use the following penalized version of the misclassification measure:

$$\hat{M}(\alpha,\beta) \equiv \frac{1}{N_\alpha N_\beta} \sum_{i\in\alpha, j\in\beta} m_{ij},$$

where N_α and N_β are the number of classes in α and β, respectively.

We are still left with the problem of actually finding good partitions. To minimize $M(\alpha, \beta)$, we can do an exhaustive search through all possible partitions. However, an exhaustive search is combinatorial in the number of classes and may be intractable when the number of classes is very large.

As an alternative to the exhaustive search, one can use a simulated annealing algorithm in which the intra-group classification measure is the energy. (See Press87) Energy state transitions then correspond to transitions between neighboring partitions where neighboring partitions are partitions that differ by an interchange of two classes. It should be noted however that it is not essential to find the optimal partition.

Once a partition has been found, the decision boundary may be further simplified by removing a class entirely from the partition and in this way defering a decision on that class until later in the tree. This removal can be performed for a class which contributes significantly to M and/or which does not significantly change M when it is moved from one subgroup to another. If the decision for a particular class is deferred, the class must be passed down both branches of the tree from the controller net.

Controller Biasing

Hybrid neural networks are subject to biasing from two sources: the local experts and the controller nets. Creating biased local experts corresponds to generating biased estimates of the decision boundaries, which is a common problem for all neural network algorithms, and can be handled with standard cross-validatory techniques. However, biasing of the controller nets corresponds to generating biased architectures and is a problem whose solution depends on the specific architecture of the hybrid network.

For tree-structure architectures, one can use CART pruning to find optimal performance (See Breiman84). Once the entire tree has been grown, we can order the set of nested sub-trees according to the following performance measure: For each node in each subtree, calculate the ratio of the performance of the subtree to the performance of a new subtree where the branches from the node of the old subtree are replaced by a single trained network. Choose the subtree with the best performance on an independent testing set.

Of course like the local experts, the controller nets are also suject to bias due to over-fitting to the training data. Therefore the controller nets should be trained using cross validation. In addition, we can try to avoid over-fitting in the controller nets by using controllers which are constrained to search for simple decision boundaries. For example, a backprop network which has only two hidden units is constrained to a much smaller family of decision boundaries than a backprop network with many hidden units. For the hybrid algorithm in this paper, constraining the controllers in this way is a desirable thing to do since the partitioning is motivated by a search for the simplest subgroup boundary.

The Algorithm

The recursive algorithm outlined below is a method for generating a tree-structured network which divides a many-class classification problem into a set of many smaller classification tasks.

1) Train a local expert to distinguish between all classes in the group.

2) Partition the group into subgroups based on the local expert's misclassification matrix.

3) Train a controller net to distinguish between subgroups.

4) Repeat steps 1) - 4) on all subgroups of three or more classes.

5) Use the CART top-down/bottom-up pruning methodology to avoid biasing.

Example

As an example, consider an imaginary ten class classification problem (Fig. 2). At the first level, the net partitions the task into one group of six classes and one group of four classes. The group of four is then be partitioned into two groups of two classes each. These groups are then passsed to local expert nets for final classification. The group of six classes is partitioned into one group of three classes and a group of four classes. Note that at this branch, the membership of one of the classes has been defered to the next branching; so class ”3” is a member of both of the class groupings made at this branch. The new group of four classes is then partitioned into two groups of two classes each, while the group of three classes is then passes directly to a local expert.

Note that the hybrid net formed is flexible to the extent that the type of networks used for the controller nets and local expert nets are not specified. They can be chosen as needed to optimize the classification performance. In addition, note that the tree-structure need not be balanced. Finally, note that the probability of correct classification for the hybrid network is the product of the probabilities of correct classification at each controller network and the final local expert.

Acknowledgements

I would like to thank Prof. Leon N Cooper and Nathan Intrator for many helpful discussions.

References

[Bachmann90] Bachmann, C. M. (1990) Learning and Generalization in Neural Networks, Ph.D. Thesis, Brown University.

[Barron89] Barron, A. R. (1989) Statistical Properties of Artificial Neural Networks, *Proc. IEEE Conf. on Decision and Control*

[Breiman84] Breiman, Friedman, Olshen and Stone (1984) Classification and Regression Trees.

[Cooper91] Cooper, L. N. (1991) Hybrid Neural Network Architectures: Equilibrium Systems That Pay Attention, *Proc. CAIP Neural Network Workshop*

[Cybenko89] Cybenko, G. (1989) Approximation by Superpositions of a Sigmoidal Function, *Mathematics of Control, Signals and Systems*

[Ersoy90] Ersoy, O. K. (1990) Parallel, Self-Organizing, Heirarchical Neural Networks, *IEEE Trans. on Neural Netowrks*

[Freidman88] Friedman, J. H. (1988) Multiple Adaptive Regression Splines, *Technical Report No. 102, Department of Statistics, Stanford University*

[Hampshire90] Hampshire, J. B. and B. A. Pearlmutter (1990) Equivalence Proofs For Multi-Layer Perceptron Classifiers and the Baysian Discriminant Function, *Proc. 1990 Connectionists Models Summer School*

[Nowlan90] Nowlan, S. J. (1990) Competing Experts: An Erperimental Investigation of Assosciative Mixture Models, *Technical Report CRG-TR-90-5, University of Toronto*

[Press87] Press, W. H., B. P. Flannery, S. A. Teukolsky and W. T. Vetterling (1987) Numerical Recipes: The Art of Scientific Computing.

[Reilly87] Reilly, D. L., C. L. Scofield, C. Elbaum and L. N Cooper (1987) Learning System Architectures Composed of Multiple Learning Modules, *Proc. IEEE First Int. Conf. on Neural Networks*

[Reilly88] Reilly, D. L., C. L. Scofield, L. N Cooper and C. Elbaum (1988) GENSEP: A Multiple Neural Network Learning System with Modifiable Network Topology *Abstracts of the First Annual International Neural Network Society Meeting*

[Sanger90] Sanger, T. D. (1990) A Tree-Structured Adaptive Network for Function Approximation in High-Dimensional Spaces, *IEEE Trans. Neural Networks*

[Sankar91] Sankar, A. and R. J. Mammone (1991) Combining Neural Networks and Decision Trees. *Proc. SPIE's Technical Conference on Apllications of Artificial Intelligence and Neural Networks*